Claims

1. (currently amended) A computer-implemented method of conformance-testing a software implementation with a software specification which defines proper behavior of the implementation, the method comprising:

applying software implementation source code and at least a portion of the software specification to produce a conformance-test enabled implementation comprising portions of the software implementation and portions of the software specification integrated into a same body of code wherein nondeterministic choices of the software specification result in assigning a corresponding choice of the conformance-test enabled implementation to a variable, wherein at least one procedure comprises at least one portion of the software implementation and at least one portion of the software specification~~;~~, and the conformance-test enabled implementation comprising a test that the variable comprises one of the nondeterministic choices of the software specification;

compiling the software implementation source code from a first high-level language into an intermediate language;

compiling the software specification from a second high-level language into the intermediate language; and

producing the conformance-test enabled implementation in the intermediate language.

2. (previously presented) The computer-implemented method of claim 1 further comprising:

the conformance-test enabled implementation comprising a test that the variable conforms to a condition on the nondeterministic choice specified in the software specification.

3. (previously presented) The computer-implemented method of claim 1 further comprising:

the conformance-test enabled implementation comprising at least one first operation to carry out when the variable comprises one of the nondeterministic choices of the software specification; and the conformance-test enabled implementation comprising at least one second operation to carry out when the variable does not comprise one of the nondeterministic choices of the software specification.

4. (canceled)

5. (previously presented) The computer-implemented method of claim 1 wherein applying the software implementation and the software specification to produce a conformance-test enabled implementation further comprises:

including in the conformance-test enabled implementation instructions of the software implementation to synchronize the state of variables of the software implementation with the state of variables of the software specification.

6. (previously presented) The computer-implemented method of claim 1 wherein applying the software implementation and the software specification to produce a conformance-test enabled implementation further comprises:

including in the conformance-test enabled implementation instructions of the software implementation to provide the choice of the conformance-test enabled implementation corresponding to the nondeterministic choice of the specification.

7. (currently amended) A computer-implemented method of conformance-testing a software implementation with a software specification, the method comprising:

producing a software object organized such that a step of the software specification is surrounded by a corresponding code section of the software implementation in the software object, the software object having a class; and

the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system;

wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object.

8. (previously presented) The computer-implemented method of claim 7 further comprising:

the software object comprising at least one instruction which, when executed by the computer system, causes a test that the state of a conformance-test enabled implementation conforms to the software specification during the mandatory call.

9. (previously presented) The computer-implemented method of claim 7, further comprising:

the conformance-test enabled implementation comprising at least one instruction which, when executed by the computer system, causes a section of the software object to be executed prior to the conformance-test enabled implementation performing the mandatory call, the section of the software object to be executed corresponding to a step of the software specification comprising the mandatory call.

10. (previously presented) The computer-implemented method of claim 7 further comprising:

modifying the software comprising the mandatory call method with instructions which, when executed by the computer system, cause instructions of the software object to be executed to test that the state of the conformance-test enabled implementation conforms to the software specification during execution of the mandatory call method.

11. (previously presented) The computer-implemented method of claim 7 wherein applying the software implementation and the software specification to produce a conformance-test enabled implementation further comprises:

including in the conformance-test enabled implementation instructions of the software implementation to synchronize the state of variables of the software implementation with the state of variables of the software specification.

12. (currently amended) An article comprising:

a machine-readable storage medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by a computer system, resulting in:

applying a software implementation source code and at least a portion of the software specification to produce the conformance-test enabled implementation comprising a same body of code with portions from both the software implementation and the software specification in the same procedure, wherein nondeterministic choices of the software specification result in assigning a corresponding choice of the conformance-test enabled implementation to a variable; and the conformance-test enabled implementation comprising a test that the variable comprises one of the nondeterministic choices of the software specification;

compiling the software implementation from a first high-level language into an intermediate language;

compiling the software specification from a second high-level language into the intermediate language; and

producing the conformance-test enabled implementation in the intermediate language.

13. (previously presented) The article of claim 12 wherein the instructions, when executed by the computer system, result in:

the conformance-test enabled implementation comprising a test that the variable conforms to a condition on the nondeterministic choice specified in the software specification.

14. (previously presented) The article of claim 12 wherein the instructions, when executed by the computer system, result in:

the conformance-test enabled implementation comprising at least one first operation to carry out when the variable comprises one of the nondeterministic choices of the software specification; and

the conformance-test enabled implementation comprising at least one second operation to carry out when the variable does not comprise one of the nondeterministic choices of the software specification.

15. (canceled)

16. (previously presented) The article of claim 12 wherein the instructions, when executed by the computer system to apply the software implementation and the software specification to produce a conformance-test enabled implementation, result in:

including in the conformance-test enabled implementation instructions of the software implementation to synchronize the state of variables of the software implementation with the state of variables of the software specification.

17. (previously presented) The article of claim 12 wherein the instructions, when executed by the computer system to apply the software implementation and the software specification to produce a conformance-test enabled implementation, result in:

including in the conformance-test enabled implementation instructions of the software implementation to provide the choice of the conformance-test enabled implementation corresponding to the nondeterministic choice of the specification.

18. (currently amended) An article comprising:

a machine-readable storage medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by a computer system, resulting in:

producing a software object organized such that a step of the software specification is surrounded by a corresponding code section of the software implementation in the software object, the software object having a class; and

the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system;

wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object.

19. (previously presented) The article of claim 18, wherein the instructions, when executed by the computer system, result in:

the software object comprising at least one instruction which, when executed by the computer system, causes a test that the state of the conformance-test enabled implementation conforms to the software specification during the mandatory call.

20. (previously presented) The article of claim 18 wherein the instructions, when executed by the computer system, result in:

the conformance-test enabled implementation comprising at least one instruction which, when executed by the computer system, causes a section of the software object to be executed prior to the conformance-test enabled implementation performing the mandatory call, the section of the software object to be executed corresponding to a step of the software specification comprising the mandatory call.

21. (previously presented) The article of claim 18 wherein the instructions, when executed by the computer system, result in:

modifying the software comprising the mandatory call method with instructions which, when executed by the computer system, cause instructions of the software object to be executed to test that the state of the conformance-test enabled implementation conforms to the software specification during execution of the mandatory call method.

22. (previously presented) The article of claim 18 wherein the instructions, when executed by the computer system, result in:

including in the conformance-test enabled implementation instructions of the software implementation to synchronize the state of variables of the software implementation with the state of variables of the software specification.

23. (previously presented) An apparatus comprising:

a processor; and

a machine-readable medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by the processor, resulting in:

applying a software implementation and the software specification to produce the conformance-test enabled implementation a same body of code with portions from both the software implementation and the software specification integrated into at least one procedure, wherein nondeterministic choices of the software specification result in assigning a corresponding choice of the conformance-test enabled implementation to a variable; and the conformance-test enabled implementation comprising a test that the variable comprises one of the nondeterministic choices of the software specification;

compiling the software implementation from a first high-level language into an intermediate language;

compiling the software specification from a second high-level language into the intermediate language; and

producing the conformance-test enabled implementation in the intermediate language.

24. (previously presented) The apparatus of claim 23 wherein the instructions, when executed by the computer system, result in:

the conformance-test enabled implementation comprising a test that the variable conforms to a condition on the nondeterministic choice specified in the software specification.

25. (previously presented) The apparatus of claim 23 wherein the instructions, when executed by the computer system, result in:

the conformance-test enabled implementation comprising at least one first operation to carry out when the variable comprises one of the nondeterministic choices of the software specification; and

the conformance-test enabled implementation comprising at least one second operation to carry out when the variable does not comprise one of the nondeterministic choices of the software specification.

26. (canceled)

27. (previously presented) The apparatus of claim 23 wherein the instructions, when executed by the computer system to apply the software implementation and the software specification to produce a conformance-test enabled implementation, result in:

including in the conformance-test enabled implementation instructions of the software implementation to synchronize the state of variables of the software implementation with the state of variables of the software specification.

28. (previously presented) The apparatus of claim 23 wherein the instructions, when executed by the computer system to apply the software implementation and the software specification to produce a conformance-test enabled implementation, result in:

including in the conformance-test enabled implementation instructions of the software implementation to provide the choice of the conformance-test enabled implementation corresponding to the nondeterministic choice of the specification.

29. (currently amended) An apparatus comprising:

a processor; and

a machine-readable medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by a computer system, resulting in:

producing a software object organized such that a series of steps of the software specification and a corresponding code section of the software implementation are enmeshed in the software object, the software object having a class;

the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system, wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object; and

the software object comprising at least one instruction which, when executed by the computer system, causes a test that the state of the conformance-test enabled implementation conforms to the software specification during the mandatory call.

30. (previously presented) The apparatus of claim 29 wherein the instructions, when executed by the computer system, result in:

the conformance-test enabled implementation comprising at least one instruction which, when executed by the computer system, causes a section of the software object to be executed prior to the conformance-test enabled implementation performing the mandatory call, the section of the software object to be executed corresponding to a step of the software specification comprising the mandatory call.

31. (previously presented) The apparatus of claim 29 wherein the instructions, when executed by the computer system, result in:

modifying the software comprising the mandatory call method with instructions which, when executed by the computer system, cause instructions of the software object to be executed to test that the state of the conformance-test enabled implementation conforms to the software specification during execution of the mandatory call method.

32. (previously presented) The apparatus of claim 29 wherein the instructions, when executed by the computer system, result in:

including in the conformance-test enabled implementation instructions of the software implementation to synchronize the state of variables of the software implementation with the state of variables of the software specification.